Algoritm de verificare a unor proprietăți pentru elemente dintr-un vector

1.Verificarea dacă toate elementele unui vector îndeplinesc o proprietate dată

a.Varianta care *nu este eficientă* parcurge toate elementele vectorului, chiar dacă este întâlnit un element care nu îndeplinește proprietatea dată

```
ok=1; //presupunem ca toate elementele indeplinesc proprietatea
for(i=0;i<n;i++)  //parcurgem toate elementele vectorului
   //daca elementul v[i] nu indeplineste proprietatea
      ok=0;   //presupunerea facuta este falsa
if(ok==1) //daca presupunerea a fost adevarata inseamna ca toate elementele indeplinesc proprietatea
  cout<<"DA"; //se adapteaza problemei de rezolvat
else cout<<"NU";
```

b.Varianta *eficientă* oprește parcurgerea elementelor vectorului dacă întâlnește un element care nu îndeplinește proprietatea dată

```
ok=1; //presupunem ca toate elementele indeplinesc proprietatea
i=0; //se pleaca de la primul element din vector
while (i<n && ok==1) //cat timp suntem in vector si elementele parcurse indeplinesc proprietatea
    //daca v[i] nu indeplineste proprietatea
      ok=0; //presupunerea facuta este falsa
   else i++; //daca elementul curent indeplineste proprietatea se trece la elementul urmator
if(ok==1) //daca presupunerea a fost adevarata inseamna ca toate elementele indeplinesc proprietatea
  cout<<"DA"; //se adapteaza problemei de rezolvat
else cout<<"NU";
```

Problema #288 – pbinfo

Se dă un şir cu n elemente, numere naturale. Să se verifice dacă toate elementele şirului sunt pare. Programul afișează pe ecran mesajul DA, dacă toate elementele şirului sunt pare, respectiv NU în caz contrar.

```
#include <iostream>
using namespace std;
int n, v[100],i,ok;
int main()
{
   cin>>n;
   for(i=0;i<n;i++)
      cin>>v[i];
  ok=1; //presupunem ca toate elementele sunt pare
i=0; //se pleaca de la primul element din vector
while (i<n && ok==1) //cat timp suntem in vector si elementele parcurse sunt pare
   if(v[i]%2==1) //daca elementul curent este impar
      ok=0; //presupunerea facuta este falsa
   else i++; //daca elementul curent este par se trece la elementul urmator
if(ok==1) //daca presupunerea a fost adevarata inseamna ca toate elementele sunt pare
  cout<<"DA";
else cout<<"NU";
return 0;
}
```

1.Verificarea dacă cel puțin un element al unui vector îndeplinește o proprietate dată

a.Varianta care *nu este eficientă* parcurge toate elementele vectorului, chiar dacă este întâlnit un element care îndeplinește proprietatea dată

```
ok=0; //presupunem ca niciun element al vectorului nu indeplineste proprietatea
for(i=0;i<n;i++)  //parcurgem toate elementele vectorului
  //daca elementul v[i] indeplineste proprietatea
     ok=1;   //am gasit un element care îndeplineste proprietatea
if(ok==1) //daca am gasit un element care indeplineste proprietatea
 cout<<"DA"; //se adapteaza problemei de rezolvat
else cout<<"NU";
```

b.Varianta *eficientă* oprește parcurgerea elementelor vectorului dacă întâlnește un element care îndeplinește proprietatea dată

```
ok=0; //presupunem ca niciun element al vectorului nu indeplineste proprietatea
i=0; //se pleaca de la primul element din vector
while (i<n && ok==0) //cat timp suntem in vector si elementele parcurse nu indeplinesc proprietatea
   //daca v[i] indeplineste proprietatea
     ok=1; //am gasit un element care indeplineste proprietatea
  else i++; //daca elementul curent nu indeplineste proprietatea se trece la elementul urmator
if(ok==1) // daca am gasit un element care indeplineste proprietatea
 cout<<"DA"; //se adapteaza problemei de rezolvat
else cout<<"NU";
```

Problema #289 – pbinfo

Se dă un şir cu n elemente, numere naturale. Să se verifice dacă în şir există elemente impare. Programul afișează pe ecran mesajul DA, dacă şirul conţine elemente impare, respectiv NU în caz contrar.

```
#include <iostream>
using namespace std;
int n, v[100],i,ok;
int main()
{
  cin>>n;
  for(i=0;i<n;i++)
    cin>>v[i];
 ok=0; //presupunem ca niciun element al vectorului nu este impar
  i=0; //se pleaca de la primul element din vector
  while (i<n && ok==0) //cat timp suntem in vector si elementele parcurse sunt pare
    if(v[i]%2==1) //daca elementul curent este impar
       ok=1; //am gasit un element care este impar
    else i++; //daca elementul curent este par se trece la elementul urmator
  if(ok==1) // daca am gasit un element impar
   cout<<"DA";
  else cout<<"NU";
return 0;
}
```

Material realizat de prof. Contraș Diana